AFRL-RQ-WP-TP-2013-0084

MODEL-BASED EVENT-TRIGGERED MULTI-VEHICLE COORDINATED TRACKING WITH COMMUNICATION DELAYS

Eloy Garcia

Infoscitex Corp.

Yongcan Cao and David W. Casbeer

Control Automation Branch
Power and Control Division

FEBRUARY 2013

STINFO COPY

AIR FORCE RESEARCH LABORATORY
AEROSPACE SYSTEMS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
February 2013	Technical Paper	01 October 2012 – 16 January 2013

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
MODEL-BASED EVENT-TRIGGERED MULTI-VEHICLE COORDINATED TRACKING WITH COMMUNICATION DELAYS	FA8650-09-D-3900-0005
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER 62201F

6. AUTHOR(S)	5d. PROJECT NUMBER
Eloy Garcia (Infoscitex Corp.) Yongcan Cao and David W. Casbeer (AFRL/RQQA)	2403
	5e. TASK NUMBER N/A
	5f. WORK UNIT NUMBER Q0X5

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)		8. PERFORMING ORGANIZATION REPORT NUMBER
Infoscitex Corp. 4025 Colonel Glenn Highway Suite 210 Dayton, OH 45431-1672	Control Automation Branch (AFRL/RQQA) Power and Control Division Air Force Research Laboratory, Aerospace Systems Directorate Wright-Patterson Air Force Base, OH 45433-7542 Air Force Materiel Command, United States Air Force	

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY ACRONYM(S)
Air Force Research Laboratory Aerospace Systems Directorate Wright-Patterson Air Force Base, OH 45433-7542 Air Force Materiel Command United States Air Force	AFRL/RQQA
	11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S) AFRL-RQ-WP-TP-2013-0084

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
PA Case Number: 88ABW-2013-0541; Clearance Date: 06 Feb 2013. This technical paper contains color.

14. ABSTRACT

This report studies the coordinated tracking problem where a group of followers intercepts a dynamic leader. It is shown in this paper that reduction of inter-agent communication is obtained and improved performance is achieved by implementing dynamical models of the leader and followers and by using an event-triggered control strategy that requires each agent to send measurement updates only when necessary. Communication delays are considered in this work and it is shown how the model-based approach can be used for time propagation of delayed measurements to obtain estimates of current positions of other agents in the network. Performance bounds on the tracking error have been obtained, which are functions of the communication topology, the event thresholds, and the time delays.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON *(Monitor)*
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	SAR	22	David W. Casbeer
					19b. TELEPHONE NUMBER *(Include Area Code)* N/A

Model-Based Event-Triggered Multi-Vehicle Coordinated Tracking with Communication Delays.

Eloy Garcia[1], Yongcan Cao[2], and David W. Casbeer[2]

[1]*Infoscitex Corp.*
Dayton, OH 45431

[2]*Control Science Center of Excellence*
Air Force Research Laboratory, Wright-Patterson AFB, OH 45433

Abstract. This report studies the coordinated tracking problem where a group of followers intercepts a dynamic leader. It is shown in this paper that reduction of inter-agent communication is obtained and improved performance is achieved by implementing dynamical models of the leader and followers and by using an event-triggered control strategy that requires each agent to send measurement updates only when necessary. Communication delays are considered in this work and it is shown how the model-based approach can be used for time propagation of delayed measurements to obtain estimates of current positions of other agents in the network. Performance bounds on the tracking error have been obtained, which are functions of the communication topology, the event thresholds, and the time delays.

I. INTRODUCTION.

Cooperative control problems have received significant attention in the last years. These problems typically require a group of mobile vehicles, also called agents, to perform a coordinated task by exchanging relevant information according to either a fixed or a time-varying communication topology. Most of the results concerning agents with continuous time dynamics require each agent to continuously transmit measurements to its neighbors [1], [2]. However, in practical problems there exist computation and, especially, communication limitations that impose constraints into how frequently an agent is able to receive measurement updates from its neighbors.

Recently, typical problems in cooperative control have been studied from a sampled-data perspective [3]-[9] where periodic samples of continuous time output signals are transmitted over a communication channel and are also used to compute sampled control inputs. The work in [3]

considers a sampled-data consensus approach for double-integrator dynamics and for undirected graphs. The authors of [4] consider sampled-data consensus, and two communication cases: synchronous and asynchronous. Reference [5] studied the consensus problem with sampled information and time-varying communication delays. Their approach considers switching communication topologies and consensus is obtained for small enough sampling periods and if there exists a frequent enough directed spanning tree in the presence of delays.

In the present paper, we extend the results on sampled-data cooperative tracking provided in [6], in order to reduce communication rates between agents and to improve performance as measured by the tracking errors. We make use of the Model-Based Event-Triggered (MB-ET) control strategy [10]-[11], [14]-[15] in which each agent implements a nominal dynamical model of other agents in the network. The states of the models provide an estimate of the real positions of neighbor vehicles and these estimated variables are used to compute the control inputs between measurement updates. Each agent generates its corresponding updates based on local error which measure the difference between real and estimated positions. The use of event-triggered control in cooperative control has been used by several authors [12], [13]. In these papers a Zero-Order-Hold (ZOH) model is used, that is, the measurement updates received by each agent are held constant until new measurements arrive. In the approach shown in this paper, the updates received from neighbors are used by each agent to update an internal model of the neighbor and provide an estimate of the neighbor's position between measurement updates.

The paper is organized as follows: Section II states the problem. Section III presents the main results of this note. Examples are provided in Section IV and conclusions are given in Section V.

II. PROBLEM STATEMENT.

The problem studied in this paper is the dynamic leader coordinated tracking problem with limited communication where a group of n agents receive information about the leader's position at some time instants and they are required to track the leader's position as close as possible. The approach we follow is a model-based approach in which each agent is equipped with dynamical models of other agents in the group, a model of the leader, and its own model as well.

Consider a group of n agents, labeled as followers. Assume the dynamics of each agent are given by the following continuous time dynamics:

$$\dot{r}_i(t) = u_i(t), \quad i = 1, 2, ..., n. \tag{1}$$

The control inputs are given in sampled-data form as follows:

$$u_i(t) = u_i[k], \quad for \ kT \le t < (k+1)T. \tag{2}$$

We use the sampling time $T>0$ to discretize the continuous time dynamics (1) to obtain:

$$r_i[k+1] = r_i[k] + Tu_i[k]. \tag{3}$$

Note that the sampling time T is only used to obtain a discrete time equivalent of the continuous time agent dynamics but it is not used to establish a periodic communication between agents. At every time k each agent will estimate the positions of other agents and its own, compute its local error, and update its local input.

The discrete time models that will be implemented by each agent are represented by:

$$\hat{r}_i^{(l)}[k+1] = \hat{r}_i^{(l)}[k] + T\hat{u}_i^{(l)}[k], \quad i = 0, 1, 2, ..., n \tag{4}$$

where $\hat{r}_i^{(l)}[k]$ represents the model state of agent i estimated by agent l and $\hat{u}_i^{(l)}[k]$ is the corresponding model control input. Note that a model of agent 0 (the leader) is also implemented by each agent. The difference between the inputs $u_i[k]$ and $\hat{u}_i[k]$ will become apparent in the following section.

In order to determine the time instants at which a given agent needs to broadcast its current measured position we use an event-triggered strategy. The events are triggered by the size of the local agent state error. The local state errors are given by:

$$\varepsilon_i^{(i)}[k] = r_i[k] - \hat{r}_i^{(i)}[k], \quad i = 0, 1, 2, ..., n. \tag{5}$$

The local errors measure the difference between the real agent position and the position estimated by its own local model. When agent i decides to transmit a measurement update then we have:

$$\hat{r}_i^{(i)}[k_\mu^i] = r_i[k_\mu^i] \tag{6}$$

where k_μ^i represents the update instants for agent i, i.e. the time instants when agent i updates its model and transmits this update; we use this notation to emphasize the fact that the updates do not take place, in general, at every time k, but at some irregular instants k_μ^i.

Each agent will transmit its current measured position to the rest of the agents if its local error is larger than a predefined threshold otherwise the agent will not attempt to send measurements

since the current estimate is close to its real position. When an agent updates its model the error is equal to zero at that time instant following the update (6). Then the following holds:

$$\left| \varepsilon_i^{(i)}[k] \right| \le \alpha \quad i = 0, 1, 2, ..., n. \tag{7}$$

III. PROPORTIONAL-LIKE CONTROLLER.

In the MB-ET framework the leader implements a model of its own dynamics. Suppose that the leader dynamics can be represented by:

$$r_0[k+1] = r_0[k] + Tu_0[k] \tag{8}$$

for unknown $u_0[k]$. Since the input for agent 0 is completely unknown, a reasonable and simple model for the leader is a ZOH model, that is, the leader will hold its latest transmitted measurement. Note that the leader only implements its own model, the ZOH model based on its broadcasted measurements. Since the leader does not receive updates from any other agent it does not need to implement models of other agents. Under these circumstances, the local state error and the ZOH model corresponding to the leader are given by:

$$\varepsilon_0^{(0)}[k] = r_0[k] - \hat{r}_0^0[k]$$
$$\hat{r}_0^0[k] = r_0[k_\mu^0], \quad for \; k_\mu^0 \le k < k_{\mu+1}^0 \tag{9}$$

where $k_\mu^0, \mu = 1, 2, ...$ represents the update instants corresponding to agent 0, that is, the time instants when the leader transmits a measurement and updates its model.

In this work we consider a communication topology that can be represented by a directed tree graph with root at the leader. A simple example of a group of agents transmitting information using this type of graph is shown in Fig. 1. In this configuration each agent i in the network does not need to implement models of every other agent, but only of those in the directed path 0-i. If agent i needs to send measurements to at least one other agent, it implements models of all agents on this directed path including itself. However, if agent i is not sending measurements to any other agent (i.e. agent i is a leaf), the agent will not implement a model of itself. For instance, agent 3 in Fig. 1 is the destination node of the directed path 0-1-3, then, agent 3 needs to implement models of agents 0, 1, and 3. Agent 6 only needs to implement models of agents 0, 1, and 3 since it never sends updates to any other agent.
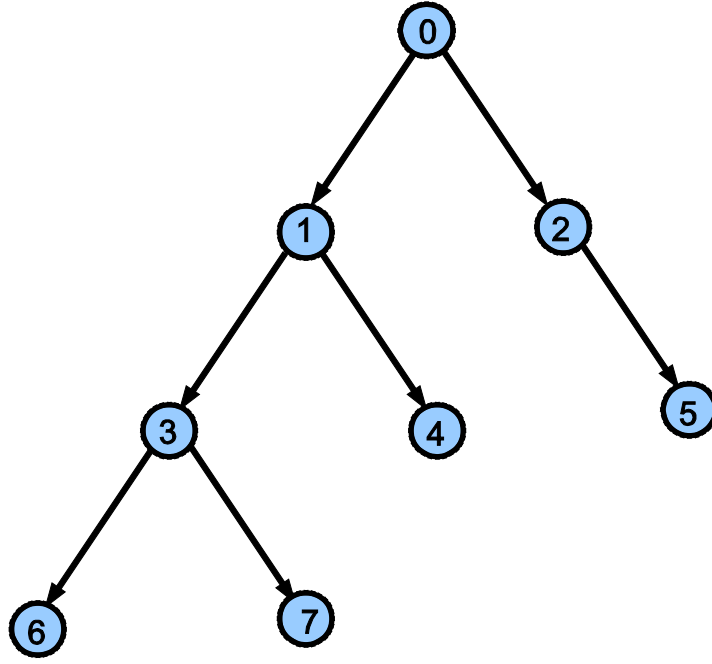
Fig. 1. A group of n followers tracking the position of a leader using a directed tree communication graph.

The control input for the *i*-th agent is given as a function of the agent's own position r_i and the estimates of the positions of the remaining agents as given by its own models. The control inputs $u_i[k]$ are given by:

$$u_i[k] = -\sum_{j=1}^{n} a_{ij}(r_i[k] - \hat{r}_j^{(i)}[k]) - a_{i0}(r_i[k] - \hat{r}_0^{(i)}[k]) \tag{10}$$

where a_{ij} is the (*i,j*) entry of the adjacency matrix associated with the follower's communication graph and $a_{i0} > 0$ if the leader is a neighbor of follower *i* and $a_{i0} = 0$ otherwise. Each agent computes its own control input since it has access to its real position $r_i[k]$.

The model control inputs used by agent *l* for each one of the agents that agent *l* is receiving information from (also including its own model control input, i.e. *l=i*) can be described by:

$$\hat{u}_i^{(l)}[k] = -\sum_{j=1}^{n} a_{ij}(\hat{r}_i^{(l)}[k] - \hat{r}_j^{(l)}[k]) - a_{i0}(\hat{r}_i^{(l)}[k] - \hat{r}_0^{(l)}[k]). \tag{11}$$

The model control inputs can be computed by all agents in the network since every agent implements the models (4) of other agents. Equations (11) are only a function of model variables; therefore, they can be computed at every node.

A. Time propagation.

The model-based approach is also useful for time propagation of delayed measurements [10]. For instance, the model implemented by agent 3 in Fig. 1 is given by:

$$\hat{r}^{(3)}[k+1] = \begin{bmatrix} \hat{r}_0^{(3)}[k+1] \\ \hat{r}_1^{(3)}[k+1] \\ \hat{r}_3^{(3)}[k+1] \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ Ta_{10} & 1-Ta_{10} & 0 \\ 0 & Ta_{31} & 1-Ta_{31} \end{bmatrix} \begin{bmatrix} \hat{r}_0^{(3)}[k] \\ \hat{r}_1^{(3)}[k] \\ \hat{r}_3^{(3)}[k] \end{bmatrix} = \hat{A}_3 \hat{r}^{(3)}[k] \tag{12}$$

which contains all models that agent 3 needs to implement, i.e. models of agents 0, 1, and 3. Suppose an update is triggered at agent 1 at time k, which means that agent 1 updates its model variables using its real position as follows:

$$\hat{r}^{(1)}[k] = \begin{bmatrix} \hat{r}_0^{(1)}[k] \\ \hat{r}_1^{(1)}[k] \end{bmatrix} = \begin{bmatrix} \hat{r}_0^{(1)}[k] \\ r_1[k] \end{bmatrix}. \tag{13}$$

The entire model state $\hat{r}^{(1)}[k]$ is transmitted by agent 1 at time k. The update (13) will be received by agent 3 after d_{31} sample delays, at time $k+d_{31}$. Instead of updating the corresponding part of its model using the delayed measurements $\hat{r}^{(1)}[k]$ agent 3 estimates the current value of $\hat{r}^{(1)}[k+d_{31}]$ using the portion of its model (12) corresponding to r_0 and r_1, that is, agent 3 computes $[\breve{r}_0^{(3)}[k+d_{31}], \breve{r}_1^{(3)}[k+d_{31}]]^T$ using the initial conditions $[\breve{r}_0^{(3)}[k], \breve{r}_1^{(3)}[k]]^T = [\hat{r}_0^{(1)}[k], \hat{r}_1^{(1)}[k]]^T$ and using the following state equation:

$$\begin{bmatrix} \breve{r}_0^{(3)}[k+1] \\ \breve{r}_1^{(3)}[k+1] \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ Ta_{10} & 1-Ta_{10} \end{bmatrix} \begin{bmatrix} \breve{r}_0^{(3)}[k] \\ \breve{r}_1^{(3)}[k] \end{bmatrix}. \tag{14}$$

Since (14) involves simple matrix operations the estimate $[\breve{r}_0^{(3)}[k+d_{31}], \breve{r}_1^{(3)}[k+d_{31}]]^T$ can be obtained instantaneously or with negligible delay after the delayed information has been received by agent 3. Then, the following update takes place:

$$\begin{bmatrix} \hat{r}_0^{(3)}[k+d_{31}] \\ \hat{r}_1^{(3)}[k+d_{31}] \\ \hat{r}_3^{(3)}[k+d_{31}] \end{bmatrix} = \begin{bmatrix} \breve{r}_0^{(3)}[k+d_{31}] \\ \breve{r}_1^{(3)}[k+d_{31}] \\ r_3[k+d_{31}] \end{bmatrix} \tag{15}$$

this means that the model variables of agent 3 are updated using the estimates of current model variables of agent 1 and the real measurement corresponding to its current local position $r_3[k+d_{31}]$. This update also generates an event and the updated model variables (15) are

6

transmitted to agents 6 and 7 in the example in Fig. 1. These agents perform the corresponding propagation operations when they receive the update at times $k + d_{31} + d_{63}$ and $k + d_{31} + d_{73}$, respectively. Agent 4 performs similar time propagations and updates as agent 3 when it receives the update from agent 1 at time $k + d_{41}$. In general, the delays are allowed to be different for each communication link.

Note that transmission events are generated when the local agent error reaches its threshold value and also when an agent receives an update from its parent such as agent 3 above; agent 3 received an update from agent 1 and it needs to resend this information to agents 6 and 7 since they also implement a model of agent 1. The vector of information is augmented to include the current position of agent 3 as well. Since the modeling error of the leader is the only uncertainty in this framework, events in the follower nodes are generated only when they receive updates from parent nodes.

Assuming that the delay d_{ij} is known to the receiving agent i, e.g. by time stamping, then we have that:

$$\breve{r}^{(i)}[k + d_{ij}] = \hat{r}^{(j)}[k + d_{ij}] \tag{16}$$

that is, the propagated variables at node i are equal to the current *model* variables at node j after delay d_{ij}. This is expected since both, model j and the propagation equation at node i, use the same parameters; then, by using the same initial conditions at time k both of them will produce the same response if the delay d_{ij} is known to agent i (and if no update has taken place at node j since the previous update). In general, and especially for long delays, we need to consider the possible updates received by node j, for the worst case delay.

B. Main Results.

Define the tracking error for each follower as follows:

$$\xi_i[k] = r_i[k] - r_0[k]. \tag{17}$$

Let $\xi[k] = [\xi_1[k], ..., \xi_n[k]]^T$ denote the tracking errors in vector form.

Proposition 1. The tracking error vector response is given by:

$$\xi[k] = Q^k \xi[0] - \sum_{l=1}^{k} Q^{k-l} T(\bar{A}_1 \varepsilon^{(1)}[l-1] + \bar{A}_2 \varepsilon^{(2)}[l-1] + ... + \bar{A}_n \varepsilon^{(n)}[l-1]) + \sum_{l=1}^{k} Q^{k-l} X_p^r[l-1] \tag{18}$$

where $Q = (I_n - TL - T\square diag\{a_{10},...,a_{n0}\})$, $\bar{A} = [A_0 \ A]$, $A_0 = [a_{10},...,a_{n0}]^T$, $\bar{A}_i$ represents the matrix obtained when all rows in $\bar{A}$ are replaced with zero row vectors except for the ith row, and $X_p^r[k] = (-r_0[k+1] + r_0[k])\mathbf{1}_n$. A and L are, respectively, the adjacency and the Laplacian matrices associated with the followers communication graph. The state model error vectors are defined by $\varepsilon^{(i)}[k] = [\varepsilon_0^{(i)}[k],...,\varepsilon_n^{(i)}[k]]^T$. Each individual error $\varepsilon_j^{(i)}[k] = r_j[k] - \hat{r}_j^{(i)}[k]$, represents the difference between the real position of an agent j compared to the model of j implemented by agent i.

Proof. The tracking error dynamics for follower i can be described by:

$$\xi_i[k+1] = r_i[k+1] - r_0[k+1]$$

$$= r_i[k] + T(-\sum_{j=1}^{n} a_{ij}(r_i[k] - \hat{r}_j^{(i)}[k]) - a_{i0}(r_i[k] - \hat{r}_0^{(i)}[k])) - r_0[k+1] \tag{19}$$

$$= \xi_i[k] - T\sum_{j=1}^{n} a_{ij}(\xi_i[k] - \xi_j[k] + \varepsilon_j^{(i)}[k]) - Ta_{i0}(\xi_i[k] + \varepsilon_0^{(i)}[k]) - (r_0[k+1] - r_0[k]).$$

The tracking errors can be expressed in compact form as follows:

$$\xi[k+1] = \begin{bmatrix} 1-T\sum_{j=0}^{n} a_{1j} & Ta_{12} & \cdots & Ta_{1n} \\ Ta_{21} & 1-T\sum_{j=0}^{n} a_{2j} & & \\ \vdots & & \ddots & \\ Ta_{n1} & & & \end{bmatrix} \xi[k] - T \begin{bmatrix} a_{10} & 0 & a_{12} & \cdots & a_{1n} \\ & & 0_{(n-1)\times(n+1)} & & \end{bmatrix} \varepsilon^{(1)}[k]$$

$$- T \begin{bmatrix} & & 0_{1\times(n+1)} & & \\ a_{20} & a_{21} & 0 & a_{23} & \cdots & a_{1n} \\ & & 0_{(n-2)\times(n+1)} & & \end{bmatrix} \varepsilon^{(2)}[k] - ... - T \begin{bmatrix} & & 0_{(n-1)\times(n+1)} & & \\ a_{n0} & a_{n1} & \cdots & a_{nn-1} & 0 \end{bmatrix} \varepsilon^{(n)}[k] + X_p^r[k] \tag{20}$$

$$= (I_n - TL - T \ diag\{a_{10},...,a_{n0}\})\xi[k] - T(\bar{A}_1\varepsilon^{(1)}[k] + \bar{A}_2\varepsilon^{(2)}[k] + ... + \bar{A}_n\varepsilon^{(n)}[k]) + X_p^r[k].$$

The response of (20) with initial conditions $\xi[0]$ and with inputs $\varepsilon^{(i)}[k], X_p^r[k]$ can be directly described as in (18).

Theorem 2. Assume that the continuous time leader's position $r_0(t)$ satisfies the following:

$$|\dot{r}_0(t)| \leq \bar{r}. \tag{21}$$

For any $T>0$, $\left| r_0(t+T) - r_0(t) \right| \leq \bar{r} T$ holds, which is equivalent to (when considering T as the discretization sampling interval)

$$\left| \frac{r_0[k] - r_0[k-1]}{T} \right| \leq \bar{r} \tag{22}$$

then, the maximum tracking error of the n followers is ultimately bounded by

$$(\bar{\varepsilon}^{(i)} + \bar{r}) \left\| (L + diag\{a_{10}, ..., a_{n0}\})^{-1} \right\|_{\infty} \tag{23}$$

when T is designed to satisfy

$$T < \min_{i=1,...,n} 1 / \sum_{j=0}^{n} a_{ij} . \tag{24}$$

Proof. From (18) the norm of the tracking error can be expressed as:

$$\left\| \xi[k] \right\|_{\infty} \leq \left\| Q^k \right\|_{\infty} \left\| \xi[0] \right\|_{\infty} + T \left\| \sum_{l=0}^{k-1} Q^l (\bar{A}_1 \varepsilon^{(1)}[l-1] + \bar{A}_2 \varepsilon^{(2)}[l-1] + ... + \bar{A}_n \varepsilon^{(n)}[l-1]) \right\|_{\infty} + T\bar{r} \left\| \sum_{l=0}^{k-1} Q^l \right\|_{\infty} \tag{25}$$

Since we consider a directed graph every *follower* receives updates from one and only one node. Then we only need to consider the effect of the estimation errors $\varepsilon_j^{(i)}[k]$ for which $a_{ij} > 0$. These errors correspond to the difference between neighbors (agent j is a neighbor of agent i).

The errors are given by:

$$\varepsilon_j^{(i)}[k] = r_j[k] - \hat{r}_j^{(i)}[k] = \varepsilon_j^{(j)}[k] + \hat{r}_j^{(j)}[k] - \hat{r}_j^{(i)}[k]. \tag{26}$$

For the first term in the right hand side of (26) we have that (7) holds. For the difference between models of agent i and its neighbor j we consider the response of the entire model state. If an event is generated by agent j at time k and the model state vector $\hat{r}^{(j)}[k]$ is sent to agent i, we are interested in the largest possible difference at time $k + \bar{d}_{ij}$, where $\bar{d}_{ij}$ represents an upper bound on the communication delay from agent j to agent i,

$$\left\| \hat{r}^{(j)}[k+\bar{d}_{ij}] - \hat{r}^{(i)}[k+\bar{d}_{ij}] \right\|_{\infty} = \left\| \hat{r}^{(j)}[k+\bar{d}_{ij}] - \breve{r}^{(i)}[k+\bar{d}_{ij}] \right\|_{\infty} \tag{27}$$

that is, the difference between the response of the model j from the time of update at time k with initial conditions $\hat{r}^{(j)}[k]$ and the time propagation of the measurement $\hat{r}^{(j)}[k]$ by agent i, where agent i assumes that no other event has been generated by agent j. In order to bound the difference (27) we consider the worst case scenario where the leader moves using its maximum

velocity and (22) holds with equality. If the delay is small such that the $\overline{rTd}_{ij} < \alpha$ holds, then we have

$$\left\| \hat{r}^{(j)}[k + \overline{d}_{ij}] - \hat{r}^{(i)}[k + \overline{d}_{ij}] \right\|_\infty \leq \left\| \hat{A}_j^{\overline{d}_{ij}} \hat{r}_\alpha^{(j)}[k] - \hat{A}_j^{\overline{d}_{ij}} \hat{r}^{(j)}[k] \right\|_\infty = \left\| \hat{A}_j^{\overline{d}_{ij}} [\alpha \ \ 0 \dots 0]^T \right\|_\infty \qquad (28)$$

where $\hat{r}_\alpha^{(j)}[k] = \hat{r}^{(j)}[k] + [\alpha \ \ 0 \dots 0]^T$ and $\hat{A}_j$ represents the all model dynamics used by agent j as illustrated in (12). In the case that the delays are larger and $\overline{rTd}_{ij} > \alpha$ holds, then additional events have been generated by agent j before the measurement $\hat{r}^{(j)}[k]$ reaches agent i. For this case we use an approximation for the response of the model of j which considers the greatest possible difference between $\hat{r}_0^{(j)}[k]$ and $\hat{r}_0^{(i)}[k]$, which is given by $\overline{rTd}_{ij} + \alpha$, as initial condition of agent j at time k. An illustration of these responses for the nodes 1 and 3 is shown in Fig. 2.
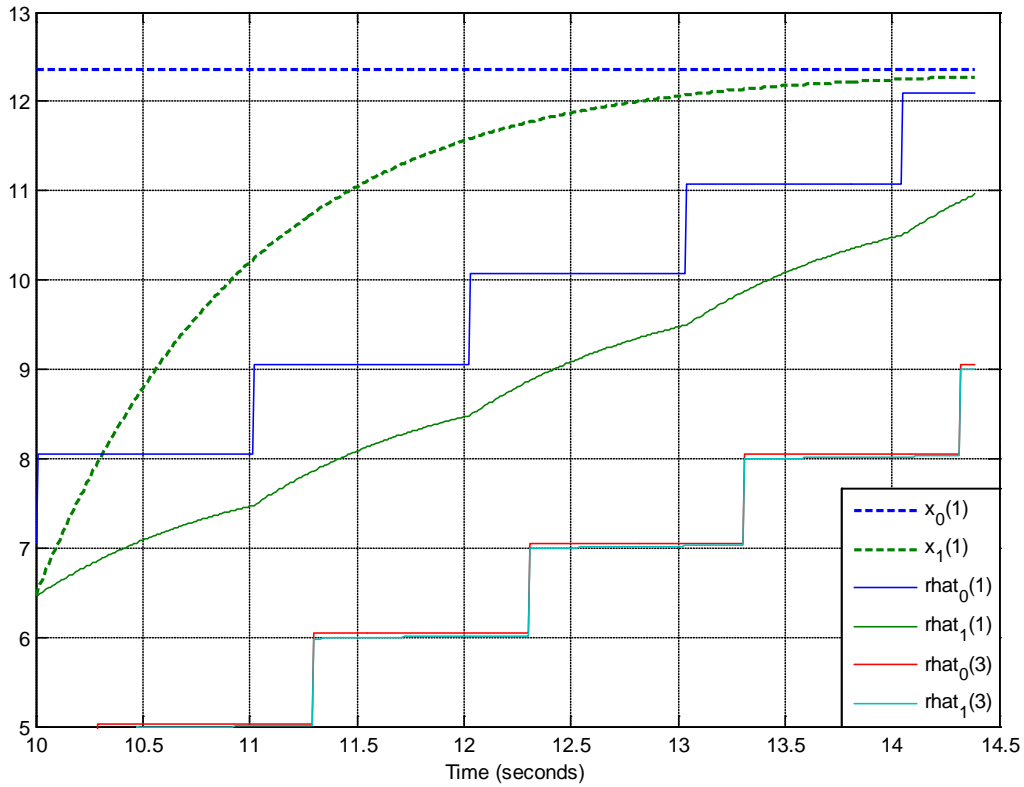

Fig. 2. Example of response of models $\hat{r}^{(1)}[k]$, $\hat{r}^{(3)}[k]$ of adjacent agents 1 and 3 with T=0.01 seconds, constant delay of 3.3 seconds and worst case leader's velocity. $x^{(1)}[k]$ represents the approximation of the response of model 1.

Figure 2 shows the approximation of $\hat{r}^{(1)}[k] = [\hat{r}_0^{(1)}[k] \ \hat{r}_1^{(1)}[k]]^T$ denoted by $x^{(1)}[k] = [x_0^{(1)}[k] \ x_1^{(1)}[k]]^T$ at $k=10$ with $T=0.01$ seconds, a constant delay of 3.3 seconds (330 sampling times) and $\alpha=1$.

In general, considering the worst case leader's velocity, we have that:

$$\left\| \hat{r}^{(j)}[k + \overline{d}_{ij}] - \hat{r}^{(i)}[k + \overline{d}_{ij}] \right\|_\infty \leq \alpha + \left\| \hat{A}_j^{\overline{d}_{ij}} [\overline{r}T\overline{d}_{ij} + \alpha \ \ 0 \ldots 0]^T \right\|_\infty \tag{29}$$

for any two agents i and j, where j is a neighbor of i. Therefore, a bound for the estimation error (26) at any time k is given by:

$$\left| \varepsilon_j^{(i)}[k] \right| \leq 2\alpha + \left\| \hat{A}_j^{\overline{d}_{ij}} [\overline{r}T\overline{d}_{ij} + \alpha \ \ 0 \ldots 0]^T \right\|_\infty \tag{30}$$

Define:

$$\overline{\varepsilon}^{(i)} = \max_{i=1\ldots n} \left| a_{ib_i} \varepsilon_{b_i}^{(i)}[k] \right| \tag{31}$$

where b_i denotes the neighbor of agent i, for $i=1\ldots n$. When all weights are equal $a = a_{ij}$ then (31) is given by:

$$\overline{\varepsilon}^{(i)} = a \max_{i=1\ldots n} \left| \varepsilon_{b_i}^{(i)}[k] \right| = a \left(2\alpha + \left\| \hat{A}_j^{\overline{d}} [\overline{r}T\overline{d} + \alpha \ \ 0 \ldots 0]^T \right\|_\infty \right) \tag{32}$$

where $\overline{d} = \max_{i=1\ldots n} \left| \overline{d}_{ij} \right|$ is the largest communication delay between any two adjacent agents.

By considering a directed graph tree we can write (20) in the following form:

$$\xi[k+1] = Q\xi[k] - T\varepsilon[k] + X_p^r[k] \tag{33}$$

where

$$\varepsilon[k] = \begin{bmatrix} a_{10} \varepsilon_0^{(1)}[k] \\ a_{2b_2} \varepsilon_{b_2}^{(2)}[k] \\ \vdots \\ a_{nb_n} \varepsilon_{b_n}^{(n)}[k] \end{bmatrix} \tag{34}$$

The response of (33) can be bounded as follows:

$$\left\| \xi[k] \right\|_\infty \leq \left\| Q^k \right\|_\infty \left\| \xi[0] \right\|_\infty + T\overline{\varepsilon}^{(i)} \left\| \sum_{l=0}^{k-1} Q^l \right\|_\infty + T r \left\| \sum_{l=0}^{k-1} Q^l \right\|_\infty \tag{35}$$

Since the leader has directed paths to all followers and $0 < T < \min\limits_{i=1,\dots,n} 1/\sum\limits_{j=0}^{n} a_{ij}$ then, by lemma 8.3 in [7], Q has all its eigenvalues within the unit circle and $\lim\limits_{k\to\infty} Q^k = 0$. Additionally, from Lemma 1.26 and Lemma 1.28 in [7], we have that:

$$\lim_{k\to\infty}\left\|\sum_{l=0}^{k-1} Q^l\right\|_{\infty} = \left\|(I_n - Q)^{-1}\right\|_{\infty} \tag{36}$$

and the maximum tracking error of the n followers, $\left\|\xi[k]\right\|_{\infty}$, is ultimately bounded by (23).

The state error for the leader contains significant model uncertainties since it is not possible to predict the behavior for this particular vehicle. A ZOH model is implemented by the followers and also by the leader itself and it was given in (9). The state error for the leader increases in general after every update for a non-stationary leader. The leader is able to store its latest broadcasted update and compute (9) in order to decide when a new update should be broadcasted to the rest of the agents. Note that (7) holds for any agent including the leader.

Remark 1. In Theorem 2 and similar to the results in [6]-[7] T has to meet certain conditions for stability, i.e. it has to be small enough so the overall system does not become unstable. Using the MB-ET framework we are able to reduce T to obtain a stable matrix Q but we do not need to communicate every T seconds. We are now able to avoid frequent communication by choosing a desired error threshold α while maintaining a stable overall system.

Remark 2. An additional advantage of the framework presented in this paper compared to the sampled-data approach [6]-[7] is that the agents are not required to transmit measurements to neighbors at the same time instants, i.e. at the sampling instants kT, $k=0,1,\dots$ In the sampled-data approach the agents are not required to communicate continuously but all of them use the same period T to sample and transmit their measurements. The communication scheme in the present paper allows for asynchronous communication in the sense that each agent determines its own transmitting time instants.

IV. EXAMPLES.

Example 1. Consider a group of $n=7$ vehicles following the position of a leader (Agent 0) and transmitting information according to the graph in Fig. 1. The parameters for this example are as

follows: $a_{10} = 1, a_{20} = 8, a_{31} = 3, a_{41} = 4, a_{52} = 5, a_{63}3 = 6, a_{73} = 6$. The sampling time is T=0.01 sec and the threshold is α=1. The delay bounds for each communication link are as follows: $\overline{d}_{10} = 2.2\,\text{sec}, \overline{d}_{20} = 2.25\,\text{sec}, \overline{d}_{31} = 3\,\text{sec}, \overline{d}_{41} = 3.05\,\text{sec}, \overline{d}_{52} = 3.1\,\text{sec}, \overline{d}_{63} = 2.5\,\text{sec}, \overline{d}_{73} = 2.9\,\text{sec}$.

Fig. 3 shows the position of the leader and followers over time (top row) and the tracking error (bottom row) for a leader with constant velocity.
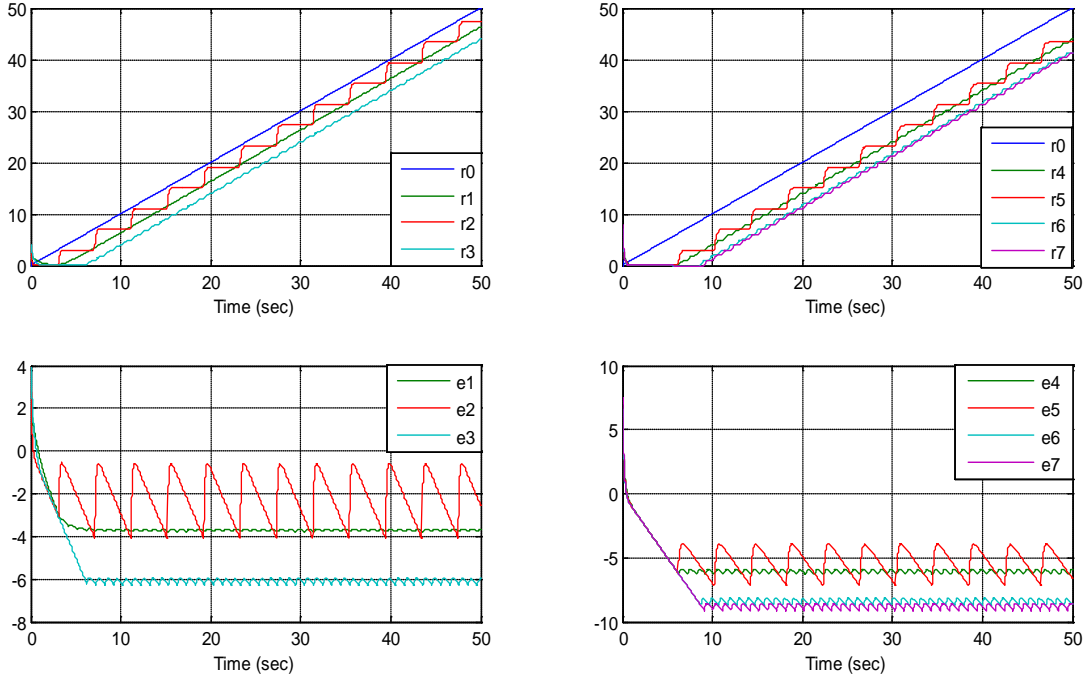

Fig. 3. Positions and tracking errors for MB-ET leader follower with constant leader velocity.

Example 2. Consider the same example with the same weights, threshold sampling time, and communication delay bounds. Fig. 4 shows the position of the vehicles over time (top row) and the tracking errors (bottom row) for the case of a leader with time-varying velocity.

V. CONCLUSIONS.

The results in this paper extend previous work concerning cooperative control and tracking of a dynamic leader. Previous results required a small enough sampling time for bounded tracking error otherwise the overall system becomes unstable. The approach in this paper implements an event-triggered control and a model-based approach that provides estimates of positions of other

agents. The overall framework is able to reduce transmission of information among agents while maintaining a bounded tracking error. Simulation examples show the effectiveness of this approach. Future work will consider the implementation of models of neighbors only (versus the entire directed path) to reduce model order and complexity, and will also consider more general communication topologies.
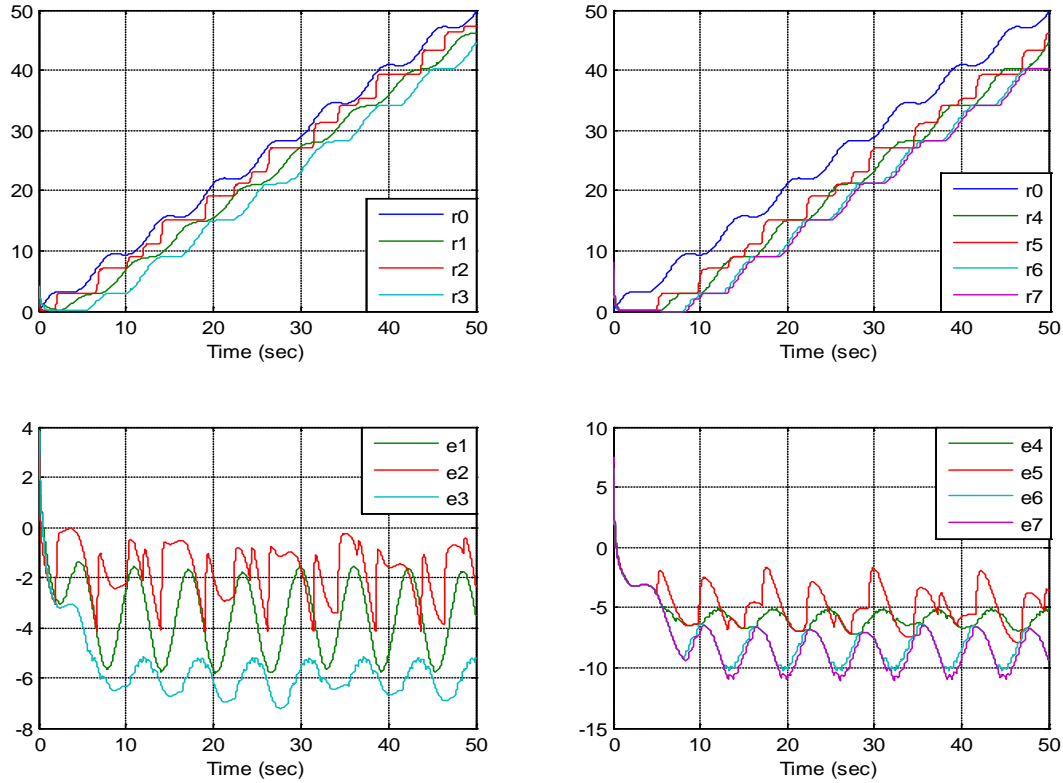

Fig. 4. Positions and tracking errors for MB-ET leader follower with time-varying leader velocity.

REFERENCES

[1] L. Moreau, "Stability of continuous-time distributed consensus algorithms," in *Proc. IEEE Conf. Decision and Control*, 2004, pp. 3998-4003.

[2] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp.71-82, 2007.

[3] T. Hayakawa, T. Matsuzawa, and S. Hara, "Formation control of multiagent systems with sampled information", in *Proc. IEEE Conf. Decision and Control*, 2006, pp.4333-4338.

[4]   Y. Gao and L. Wang, "Consensus of multiple double integrator agents with intermittent measurements," *International Journal of Robust Nonlinear Control*, vol. 20, no. 10, pp. 1140-1155, 2010.

[5]   J. Qin and H. Gao, "A sufficient condition for convergence of sampled-data consensus for double-integrator dynamics with nonuniform and time-varying communication delays," *IEEE Transactions on Automatic Control*, early access, 2012.

[6]   Y. Cao, W. Ren, and Y. Li, "Distributed discrete-time coordinated tracking with time-varying reference state and limited communication," *Automatica*, vol. 45, pp. 1299-1305, 2009.

[7]   W. Ren and Y. Cao, "Distributed coordination of multi-agent networks: emergent problems, models, and issues," Springer-Verlag, London, 2011.

[8]   Y. Cao and W. Ren, "Sample-data formation control under dynamic directed interaction," in *Proc. American Control Conf.*, 2009, pp. 5186-5191.

[9]   Y. Cao and W. Ren, "Multi-vehicle coordination for double-integrator dynamics under fixed undirected/directed interaction in a sampled data setting," *International Journal of Robust Nonlinear Control*, vol. 20, no. 9, pp. 987-1000, 2010.

[10] E. Garcia and P. J. Antsaklis, "Model-based event-triggered control with time-varying network delays," in *Proc. IEEE Conf. Decision and Control-European Control Conference*, 2011, pp. 1650-1655.

[11] E. Garcia and P. J. Antsaklis, "Model-based event-triggered control for systems with quantization and time-varying network delays," *IEEE Transactions on Automatic Control*, vol. 58, no.2, pp. 422-434, 2013.

[12] D. V. Dimarogonas and K. H. Johansson, "Event-triggered control of multi-agent systems," *Joint IEEE Conf. Decision and Control and 28th Chinese Control Conference*, 2009, pp. 7131-7136.

[13] H. Yu and P. J. Antsaklis, "Quantized output synchronization of networked passive systems with event-driven communication," in *Proc. American Control Conf.*, 2012, pp. 5706-5711.

[14] E. Garcia and P.J. Antsaklis, "Decentralized model-based event-triggered control of networked systems," in *Proc. American Control Conference,* 2012, pp. 6485-6490.

[15] Y. Sun, and N. H. El-Farra, "Dynamic quasi-decentralized control of networked process systems with limited measurements," in *Proc. American Control Conference,* 2011, pp. 3152-3157.